SCHEME OF TEACHING AND EXAMINATION
B.E. (ISE) III SEMESTER

Sl No.	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UMA301C	Mathematics-III	04	4	0	0	50	50	100
2	UIS302C	Discrete Mathematics & Graph Theory	04	4	0	0	50	50	100
3	UIS303C	Logic Design	04	4	0	0	50	50	100
4	UIS304C	Object Oriented Programming with C++	04	4	0	0	50	50	100
5	UIS305C	Data Structures Using C	04	4	0	0	50	50	100
6	UIS306L	Object Oriented Programming Laboratory	1.5	0	0	3	50	50	100
7	UIS307L	Logic Design & Computer Hardware Laboratory	1.5	0	0	3	50	50	100
8	UIS308L	Data Structures Laboratory	1.5	0	1	2	50	50	100
9	UMA300M	Advanced Mathematics-I	--	4	--	--	50	50	100*
		Total	**24.5**	**20****	**1**	**8**	**400**	**400**	**800**

* Advanced Mathematics-I is a mandatory subject only for students having Diploma and admitted to 3rd Semester through lateral entry scheme. Passing the subject is compulsory: however marks will not be considered for awarding grade/class. A PP/NP grade will be awarded for passing/not passing the subject.

** The total lecture hours for students having Diploma and admitted to 3rd Semester through lateral entry scheme is 24 hours.

SCHEME OF TEACHING AND EXAMINATION
B.E. (ISE) IV SEMESTER

Sl No.	Subject Code	Subject	Credits	Hours/Week			Examination Marks		
				Lecture	Tutorial	Practical	CIE	SEE	Total
1	UMA401C	Mathematics-IV	04	4	0	0	50	50	100
2	UIS402C	Microprocessors	04	4	0	0	50	50	100
3	UIS403C	Analysis and Design of Algorithms	04	4	0	0	50	50	100
4	UIS404C	Computer Organization	04	4	0	0	50	50	100
5	UIS405C	File and Information Structures	04	4	0	0	50	50	100
6	UIS406C	Theoretical Foundations of Computer Science	03	3	0	0	50	50	100
7	UIS407L	File and Information Structures Laboratory	1.5	0	0	3	50	50	100
8	UIS408L	Analysis of Algorithms Laboratory	1.5	0	0	3	50	50	100
9	UMA400M	Advanced Mathematics-II	--	4	--	--	50	50	100*
		Total	**26**	**23****	**0**	**6**	**400**	**400**	**800**

* Advanced Mathematics-II is a mandatory subject for students having Diploma and admitted to 3rd Semester through lateral entry scheme. Passing the subject is compulsory: however marks will not be considered for awarding grade/class. A PP/NP grade will be awarded for passing/not passing the subject.

** The total lecture hours for students having Diploma and admitted to 3rd Semester through lateral entry scheme is 27 hours.

III SEMESTER

ADVANCED MATHEMATICS-I
(Common to all branches)

Sub Code: DMA301M **IA Marks:**
Hrs/Week: 04 **Exam Hours: 03**
Total Hrs.: 40 **Exam Marks:**

1) **Trigonometry:** Complex Numbers : Definitions, complex numbers as an ordered pair, real and imaginary parts, modulus and amplitude of a complex number, equality of a complex number, addition, subtraction, multiplication & division of complex numbers, polar form, Argand Diagram, exponential form, expressing in the form a + ib problems.
06 Hours

2) **Differential Calculus:** Differentiation of nth order of standard functions, Leibnitz's theorem (Statement only) with examples, polar curves, Taylor's series, Maclaurian's series of simple functions for single variable Partial Differentiation: Definition, Euler's theorem, total differentiation, Differentiation of composite and implicit functions, jacobians illustrative examples and problems
14 Hours

3) **Integral Calculus:** Reduction formula for functions $Sin^n x$, $Cos^n x$, $Sin^m x\ Cos^n x$. Double integral, simple problems & Triple integral simple problem(with standard limits), β and γ functions, properties, relation between β and γ functions simple problems
08 Hours

4) **Differential Equations:** Solution of first order, first degree differential equations – variable separable methods, homogeneous equation, Bernoulli's and exact differential equations (without I.F.) Differential equations of second and higher orders with constant coefficients.
12 Hours

Text Books:

1) Higher Engineering Mathematics – B S Grewal
2) Higher Engineering Mathematics – H K Dass

III SEMESTER

ENGINEERING MATHEMATICS - III

Sub Code: UMA301C **IA Marks:**
Hrs/Week: 04 **Exam Hours: 03**
Total Hrs.: 52 **Exam Marks:**

Unit-I

NUMERICAL ANALYSIS:

Roots *of* Transcendental equations using Bisection Method, Regula-Falsi Method and Newton-Raphson. Method, Finite differences, forward, backward and central difference operators (no derivations on relations between operators), Newton-Gregory forward and backward interpolation formule. Stirling's and Bessel's interpolation formulae (without proof), Lagrange's and Newton's divided difference interpolation formulae (without proof), Inverse interpolation using Lagrange's interpolation formula, Numerical differentiation using Newton's forward & backward formulae.

Numerical integration-Trapezoidal rule, Simpson's one third rule, Simpson's three eighth rule and Weddle's rule (no derivation *of* any formulae). Numerical solutions *of* first order ODE - Taylor's series Method, Modified Euler's method, Runge-Kutta 4th order method, Milne's Predictor and *Corrector* method (problems only). **14 Hours**

Unit-II

FOURIER SERIES, FOURIER TRANSFORMS, Z- TRANSFORMS:

Periodic functions, Conditions for Fourier series expansions, Fourier series expansion *of* continuous functions and functions having infinite number *of* discontinuities, even and odd functions. Half-range series, Practical Harmonic Analysis.

Infinite Fourier transforms and inverse Fourier transforms- simple properties, Complex Fourier transform, Fourier sine and Fourier cosine transforms, Inverse Fourier sine & cosine transforms, Convolution theorem & Parseval's identity (without proofs).

Z - transform - definition, standard forms, Linearity property, damping rule, shifting rule - problems. **14 Hours**

Unit-III

PARTIAL DIFFERENTIAL EQUATIONS:

Formation *of* partial differential equations by elimination *of* arbitrary constants and arbitrary functions, Solution *of* equation *of* the type: Pp+Qp=R, Charpit's method, Solution *of* PDEs by the method *of* separation *of* variables.

Derivation *of* one-dimensional heat and wave equations. Numerical solutions *of* One-dimensional heat and wave equations by explicit method, Laplace equation by using standard five points formula. **12 Hours**

Unit-IV

LINEAR ALGEBRA:

Rank *of* a matrix by elementary row transformations, Consistency *of* system *of* linear equations, Gauss elimination method, Gauss - Seidel method, characteristic values and characteristic vectors *of* matrices (no theorems), Largest Eigen value and the corresponding Eigen vector by Power Method.

CALCULUS OF VARIATIONS:

Variation *of* a function and a functional, Extremal *of* a functional, Variational Problems, Euler's equation, Standard variational problems including geodesics, minimal surface *of* revolution, hanging chain and Brachistochrone problems. **12 Hours**

TEXT BOOKS: Higher Engineering Mathematics (36th edition-20OZ) by Dr. B S Grewal, Khanna Publishers, New Delhi.

Unit-I: **Numerical Analysis:**

Chapter 24: 24.1,
Chapter 25:25.1, 25.5 to 25.7, 25.12 to 25.14, 25.16
Chapter 27:27.1,27.3 to 27.5 27.7, 27.8

Unit-II: **Fourier series Fourier Transforms and Z Transforms:**

Chapter 10: 10.1 to 10.7, 10.11
Chapter 22: 22.1, 22.4 to 22.7
Chapter 26: 26.9 to 26.14

Unit-III: **Partial Differential Equations:**

Chapter 17: 17.1 to 17.3, 17.5,17.7
Chapter 18: 18.1 to 18.4 (1), 18.5(1)
Chapter 28: 28.1 to 28.5

Unit-IV: **Linear algebra and Calculus of variations:**

Chapter 2: 2.9(1), 2.9(2), 2.11(1), 2.14(2)
Chapter 24:24.4, 24.5(1), 24.6(2), 24.8
Chapter 30: 30.1 to 30.5

REFERENCE BOOKS: Advanced Engineering Mathematics by E Kreyszig (John Wiley & Sons)

Unit-I	Roots of Transcendental equations-Numerical different ion Numerical Integration	One Question One Question
Unit-II	Fourier series Fourier Transforms & Z-transforms	One Question One Question
Unit-III	Formation of PDE's, Method of separation variables Derivation of One dimensional Heat and wave equation	One Question One Question
Unit-IV	Linear Algebra Calculus of variations	One Question One Question

Discrete Mathematics and Graph Theory

Semester	**: III**	**Credits**	**: 4**
Code	**: UIS302C**	**Teaching hours/week**	**: 4**
		Total Teaching Hours	**: 52**

UNIT I

Fundamentals: Sets and Subsets, Operations on Sets, Sequences, Division in the Integers, Matrices, Mathematical Structures.

Logic: Prepositions and Logical Operations, Conditional Statements, Methods of proof, Mathematical Induction. Counting: Permutations, Combinations, Pigeonhole Principle, Elements of Probability, Recursive Relations. **12 Hrs.**

UNIT II

Relations: Product sets and Partitions, Relations and Diagraphs, Paths in Relations and Diagraphs, Properties of relations, Equivalence relations, Computer representations of Relations and Diagraphs, Operations on relations, Transitive closure and Warshell's algorithm

Functions: Functions, Functions for Computer Science, Growth of functions, Permutation Functions. Order Relations: Partially Ordered Sets, External Elements of Partially Ordered Sets, Lattices.

Semigroups and Groups: Semigroups, Products and quotients of Semigroups, Groups, Products and quotients of Groups, Other Mathematical Structures. **14 Hrs.**

UNIT III

Introduction: What is a graph? Applications of Graphs, Finite and Infinite Graphs, Incidence and Degree, Isolated vertex, Pendent Vertex and Null Graphs.

Paths and Circuits: Isomorphism, Subgraphs, Walks, Paths, Circuits, Connected Graphs, Disconnected Graphs, Components, Euler Graphs, Operations on graphs, Hamiltonian Paths and Circuits, Traveling salesman Problem.

Trees and Fundamental Circuits: trees, Properties of trees, Pendent vertices in trees, Distance and centers in trees, Rooted and Binary trees, Counting trees, Spanning trees, Fundamentals circuits, Finding all Spanning trees of a graph, Spanning trees in weighted Graph. **12 Hrs.**

UNIT IV

Cuts and Vertices: Cut Sets, Properties of Cut sets, All Cut Sets in a Graph, Fundamental Circuits and cut sets, Connectivity and separability, Network flows,1-isomorphism, 2-isomorphism.

Planar Graphs: Planar Graphs, Representation and detection of planar graphs.

Matrix representation of Graphs: Incident Matrix, Submatrices of A(G). Circuit matrices, Fundamental circuit matrix and Rank of B, An application to switching network, Cut-Set matrix, Relationships among A_f, B_f, and C_f, Path Matrix, Adjacency Matrix.

Coloring, Covering and Partitioning: Chromatic Number, Chromatic Partitioning, Chromatic Polynomial.

Directed Graphs: What is a directed Graph? Types of Directed graphs, Digraphs and Binary Relations, Directed Paths and connectedness, Euler Digraphs, Trees with directed edges, Fundamental Circuits in directed Graphs, Adjacency Matrix of Digraphs **14 Hrs.**

Text Books:

1. Kolman, Busby, Ross, "Discrete Mathematical Structures", 5th Edition, PHI. (Chapters 1, 2, 3, 4, 5, 6.1, 6.2, 6.3, 9)
2. Narasingh Deo, "Graph Theory with Applications to Engineering and Computer Science", PHI. (Chapters 1, 2, 3, 4, 5.1, 5.2, 5.3, 5.4, 7, 8.1, 8.2, 8.3, 9)

Reference Books:

1. Gary Haggard, John Schlipf, Sue Whitesides, "Discrete Mathematics for Computer Science", Thomson Publications.
2. Ralph P Grimaldi, "Discrete and Combinatorial Mathematics – An Applied Introduction", Pearson Education, 4th Edition.

LOGIC DESIGN

Semester : III **Credits:4**
Code : UIS303C **Teaching hours/week: 4**
Total Teaching Hours: 52

UNIT I

Number Systems, Arithmetic and Codes
Positional number system, Counting in a positional number systems, Representation of signed numbers, Binary coded decimal, Codes

Boolean algebra and Combinational networks
Definition of Boolean algebra, Principle of duality, Boolean formulas and functions, Canonical formulas, Manipulation of Boolean formulas and combinational circuits, Incomplete Boolean functions and Don't care conditions, Additional Boolean operations and gates, Introduction to HDL

12 Hrs

UNIT II

Simplification of Boolean expressions
Formulation of the Simplification problem, Prime implicants and Irredundant Disjunctive expressions, Prime implicates and Irredundant conjunctive expressions, Karnaugh maps, Using K-map to obtain minimal expression, Minimal expressions for incomplete functions, The Quine-McClauskey method of generating prime implicants and prime implicates, Prime implicants / prime implicates tables and Irredundant expressions, Decimal method for obtaining prime implicants, Variable Entered K-map, HDL implementation of logic circuits

Logic Design with MSI components
Binary adders and subtracters, Decimal Adders, Comparators, Decoders, Encoders, Multiplexers

14 Hrs

UNIT III

Programmable logic Devices
Programmable Logic Devices, PROM, Programmable Logic Array, Programmable Array Logic, HDL implementation.

Flip-Flops and applications
Basic bistable element, Latches, Master Slave Flip-Flops, Edge Triggered Flip-Flops, Characteristic equations, Registers, Counters, Design of Synchronous Counters, HDL implementation

14 Hrs

UNIT IV

Synchronous Sequential networks

Structure and operation of clocked Synchronous sequential networks, Analysis of clocked Synchronous sequential networks

D/A Conversion and A/D conversion

Variable, resistor networks, Binary Ladders, D/A converters, D/A accuracy and resolution, A/D converter- Simultaneous conversion, A/D converter – Counter Method, Continuous A/D conversion, A/D techniques ,Dual slope A/D conversion ,A/D accuracy and resolution **14 Hrs**

Text Books.

1. Digital Principles and design- Donald D. Givone (TMH)(Chapters 2.1, 2.2, 2.7, 2.10, 3.1.1, 3.4, 3.5, 3.6, 3.7, 3.8, 3.8, 4.1, 4.2, 4.3, 4.5, 4.6, 4.7.2, 4.8, 4.9.2, 4.11, 5, 6.1, 6.2, 6.4, 6.5.1, 6.5.2, 6.6, 6.7, 6.8, 6.9, 7.1,7.2)
2. Digital Principles and applications – Albert Paul Malvino, Donald P Leach and Goutam Saha (TMH), 6^{th} Edition (chapters 2.5, 3.11, 4.14, 6.128.12, 9.7, 10.9, 12)

Reference Books

1. Fundamentals of Digital Logic with Verilog Design, Stephen Brown, Zvonko Vranesic, TMH, 2006
2. Fundamentals of Logic Design, Charles H. Roth, Jr., 5th Edition,Thomson, 2004
1. Digital Systems Principles and Applications, Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, 10th Edition, PHI/Pearson Education, 2007

Object Oriented Programming With C++

Sem	**: III**	**Credits**	**: 4**
Sub code	**: UIS304C**	**Hrs/week**	**: 4**
		Total hrs	**: 52**

UNIT I

Principles of Object Oriented Programming: Software Evolution, A look at procedure-oriented Programming, Object-oriented programming paradigm, Basic concepts of OOP, Object Oriented languages, Applications of OOP.

Introduction to C++: Console Input/Output in C++, Variables in C++, Reference Variables in C++, Function Prototyping, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions

Classes and Objects: Introduction to Classes and Objects, Member Functions and Member Data, Objects and Functions, Objects and Arrays, Namespaces, Nested Classes

12 Hours

UNIT II

Constructors and Destructors: Constructors, Destructors, the Philosophy of OOPS

Inheritance: Introduction to Inheritance, Base Class and Derived Class Pointers, Function Overriding, Base Class Initialization, The Protected Access Specifier, Deriving by Different Access specifiers, Different kinds of Inheritance, Order of invocation of constructors and destructors.

Dynamic Memory Management: Introduction, Dynamic Memory Allocation, Dynamic Memory Deallocation

13 Hours

UNIT III

Operator Overloading: Operator Overloading, Overloading the Various Operators - Overloading the Increment and the Decrement Operators (Prefix and Postfix), Overloading the Unary Minus and the Unary Plus Operator, Overloading the Arithmetic Operators, Overloading the Relational Operators , Overloading the Assignment Operator, Overloading the Insertion and Extraction Operators, Overloading the new and the delete Operators, Overloading the Subscript Operator, Overloading the Pointer-to-member Operator (->)

13 Hours

UNIT IV

Virtual Functions and Dynamic Polymorphism: The Need for Virtual Functions, Virtual Functions, The Mechanism of Virtual Functions, Pure Virtual Functions, Virtual Destructors and Virtual Constructors

Templates: Introduction, Function Templates, Class Templates, The Standard Template Library (STL)

Stream Handling: Streams, The Class Hierarchy of Handling Streams, Text and Binary Input/Output, Text Versus Binary Files, Text Input/Output, Binary Input/Output, Stream Handling contd.: Opening and Closing Files, Files as Objects of the fstream Class

Type Conversion: Type Conversion, New Style Casts, and RTTI

Exception Handling: Introduction, C-Style Handling of Error-generating Codes, C++ Style Solution - the try/throw/catch Construct, Limitation of Exception Handling

14 hours

TEXT BOOKS

1. Object-Oriented Programming with C++, Sourav Sahay, Oxford University Press, 2006 Chapter 1 (1.5 - 1.11), Chapter 2, 3(3.1 – 3.3), 4, 5, 6, 7 (7.1 – 7.8), 8, 9, 10

2. Object-Oriented Programming with C++ by E. Balaguruswamy. Tata McGraw Hill - II Edition Chapter 1(1.2 - -1.8)

REFERENCE BOOKS

1. The Complete Reference C++ by Herbert Schildt, Tata McGraw Hill - IV Edition

2. Object-Oriented Programming with C++ by P.B.Kotur

Data structures using C

Sem : III **Credits : 4**
Sub code : UIS305C **Hrs/week : 4**
Total hrs : 52

Unit I

Structures & Unions: Instances of structures.Declaration of structure. Scope of structure variables. Accessing members of structures, size of structure, Assigning values of one structure variable to another. Initialization of structure variable, Array of structure, Nested Structures. Structures & Functions. Union. **5 hrs**

Pointers & their usages: Pointer Declaration. Pointer Arithmetic, Pointer & arrays, Pointers & structures, Arrays of pointers, Function pointers. **Strings:** Character arrays, Sample programs, two dimensional arrays character arrays**.** **3 hrs**

Files in C: File handling in C by using File pointers, Additional functions for file handling, use of file handles to access & manipulate files. **2 hrs**

Advanced concepts: Enumerated data types, User defined data types, Storage Classes, Bite wise operators, preprocessor directives & Macro definitions, Command line arguments.

3 hrs

Unit II

The stack: ***Definition and Examples:*** Primitive operations, The stack as an Abstract data type, ***Representing Stacks in C***: Implementing pop operation, Testing for Exceptional Conditions, Implementing the push operations. ***An Example- Infix, Postfix and Prefix*** : Evaluating a postfix Expression, Program to evaluate a postfix Expression, Limitations of the program, converting an Expression from Infix to Postfix, program to convert an Expression from Infix to Postfix. **6 hrs**

Recursion: ***Recursive definition and processes***: The factorial function, Multiplication of Natural numbers, The Fibonacci sequence, The Binary search, properties of recursive definitions or Algorithms. ***Recursion in c:*** Factorial in C, The binary search in C, Recursive chains, Recursive

definitions of algorithmic Expressions, ***Rewriting recursive programs***: The Towers of Hanoi Problem, Translation from Prefix to Postfix Using Recursion, ***Simulating recursion***: Return from a function, Implementing recursive functions, simulation of factorial, Improving the simulated routine, Eliminating goto's, simulating the tower of Honai. ***Efficiency of recursion***. **4 hrs**

Queues: ***The queue and its sequential representation***: The queue as an abstract data type, C implementation of queue, The insert operation, The priority queue, Array implementation of a priority queue**.** **3 hrs**

Unit III

Lists: ***Linked lists***: Inserting and removing nodes from a list, linked implementation of stacks, the getnode and freenode operations, linked implementation of queues, the linked list as a data structure, examples of list operations, list implementation of priority queues, ***lists in c***: array implementation of lists, limitations of the array implementation, allocating and freeing dynamic variables, linked lists using dynamic variables, queues as lists in c, non integer and nonhomogeneous lists, comparing the dynamic and array implementation of lists, implementing header nodes. ***An Example-simulating using linked lists:*** the simulation process, data structures, and the simulation program. **7 hrs**

Other list structures: Circular lists, the stack as a circular lists, the queue as a circular list, primitive operations on circular lists, the Josephus problem, header nodes, addition of long positive integers using circular lists, doubly linked lists, addition of long integers using doubly linked lists**.**

6 hrs

Unit IV

Trees: ***Binary trees:*** operation on Binary trees, Applications of Binary Tree, ***Binary tree representation****:* Node representations of Binary Trees, Internal & External Nodes, Implicit array representation of binary trees. Choosing a binary tree representation, binary traversal in c. Threaded Binary trees. Traversal using father field, Heterogeneous Binary Trees. ***An Example- The Huffman Algorithm:*** The Huffman algorithm, A c program, ***Representing Lists as Binary trees:*** Finding the kth element, deleting an element, implementing tree represented lists in C, Constructing a tree

represented list, The Josephus problem revisited, ***Trees and their applications*** : C representation of trees, tree traversals, general expressions as trees, evaluating an expression tree, constructing tree, ***An example – Game trees.*** **8 hrs**

Sorting: ***General background***: Efficiency considerations, O notation, efficiency of sorting, ***Selection and tree sorting***: straight selection sort.

Searching: ***Basic search techniques:*** the dictionary as an Abstract data type, algorithm notation, Sequential searching, Efficiency of sequential searching, Recording a list for maximum search efficiency, searching an ordered table, the indexed sequential search, the binary search, interpolation search. **5 hrs**

Text books:

1) Basavraj S Anami,Shanmukhappa A Angadi. Sunilkumar S.Manvi."Computer Concepts and C Programming " **A Holistic Approach to learning C**. PHI Publications 2007. Chapters (9,10, 7.3,11,13).
2) Aaron M. Tennenbaum, Yedidyah Langsam and Moshe J. Augenstein, "Data structure using C ", Pearson Education/PHI 2006 (chatpter2, 3, 4, 5, 6, and 7)

Reference Books:

1) Behrouz A. Forouzan and Richard F. Gilberg, Thomson, "Computer Science A structured Programming Approach usig C " II edition, 2003.
2) Richard F. Gilberg and Behrouz, Data structures "A pseudo code approach with c " , Thomson, 2005.
3) Robert Kruse and Breuse Leung," Data structures and program Design in C", PEARSON Education, 2007

OBJECT ORIENTED PROGRAMMING LABORATORY

Sem	**: III**	**Hrs/Week**	**: 3Hrs**
Sub Code	**: UIS306L**	**Credits**	**: 1.5**

1. Program using array of class objects

   Create a class STUDENT with data members rollno, name and percentage. Create 'n' number of objects of class STUDENT and initialize the data members of each object and display them.

2. Program using constructors and destructors

   Write a C++ program to create an object of class BOOK with title of the book, author, publisher and price as its data members. Illustrate the constructor and destructor functions associated with the BOOK object.

3. Program using inheritance

   Write a C++ program to illustrate multilevel inheritance.

4. Program using inheritance

   Write a C++ program to illustrate hybrid inheritance using hierarchical-multiple inheritance.

5. Program using function overloading

   Write a C++ program to overload a member function search () to search an integer key value and a key value of type double.

6. Program using function overriding

   Write a C++ program to illustrate function overriding.

7. Program using Operator overloading

   Write a C++ program to define a class called COMPLEX containing two members real and imaginary and illustrate how the binary operators +,-,* and / can be overloaded for the objects of type COMPLEX.

8. Program using Operator overloading

   Write a C++ program to define a class called STACK and implement its operations using overloaded increment and decrement operators.

9. Program using Operator overloading

   Write a C++ program to define a class called STRING containing two strings and overload operator + for concatenating new and delete operators for allocating and deallocating memory for STRING objects. Display the result after concatenation by overriding the operator <<.

10. Program on virtual functions

    Write a C++ program to demonstrate virtual functions.

11. Program using I/O Streams

    Write a C++ program to create a file called Source.txt and print its contents to the file Destin.txt so that the contents in Destin.txt are reverse of Source.txt.

12. Program using i/o Streams

    Write a C++ program to create a text file and determine whether the end of file condition is encountered or not.

13. Program using templates

    Write a C++ program to create a template class QUEUE with add, delete and display member functions. Using this class template, implement a queue of integers and doubles.

14. Program on binary tree

    Write a C++ program to create a class called BIN_TREE(Binary Tree) with member functions to perform inorder, preorder and postorder traversals. Create a BIN_TREE object and demonstrate the traversals.

15. Program on Exception Handling

    Write a C++ program to illustrate Exception Handling by making use of try, throw and catch constructs.

Logic Design and Computer Hardware Laboratory

Sem	**: III**	**Hrs/Week**	**: 3Hrs**
Sub Code	**: UIS307L**	**Credits**	**: 1.5**

Logic Design Lab

1. Realization of a given Boolean Expression using MEV method.
2. Design and implementation of BCD to Excess-3 using 4-Bit Adder Chip and Logic Gates.
3. Design and implement Full adder using 3:8 Decoder (74138) and Simulate full adder.
4. Design ad implement Full subtracter using 8:1 multiplexer (74154) and simulate the 8:1 Multiplexer.
5. Design JK master/slave FF using NAND gates and simulate.
6. Design and implementation of 3 bit Mod-n synchronous counter using JK flip-Flops(7476) (where n<=8) and simulate
7. Design and implementation of Ring counter and Johnson counter using 4-bit shift register and simulate
8. Design and implementation of an Asynchronous Counter using a Decade Counter IC to count up from 0 to n (n<9). Display the count value on 7-segment LED display using BCD to 7-segment code converter IC.
9. Design of 4-bit R-2R ladder DAC using Op-Amp.

Hardware lab

1. PC Assembly & Dis-assembly
2. Demonstration of Bus operation
3. Storage Device swapping.
4. Parallel port, serial port and USB port testing
5. Understand BIOS and CMOS
6. Understand Chipset
7. Backup and Restore techniques
8. Data Recovery techniques

Reference Books

1. Trouble shooting your PC --- M. David stone and alfred poor PHI edition
2. Enhanced guide to managing and maintaining your PC Jean Andrews, Thomson course technology, III edition
3. IBM PC & CLONES Fifth Edition EBPB - Rajesh Hongal

DATA STRUCTURE LAB

Sem : III **Hrs/Week : 3Hrs**
Sub Code : UIS308L **Credits : 1.5**

Programs on structures and unions:

1) Consider a structure to specify data on students stored in an array of N students given below: Roll number, Name, Department, Customer, Year of passing.
    a) Write a function to print names of all students who joined in a particular year
    b) Write a function to print the data of a student whose roll number is given.
2) Write a C program by using unions to represent a student's accomodation details, which are to be for a house if the student is a local and for a hostel otherwise.

Programs on pointers and their usage:

3) Write a C program to compute roots of a quadratic equation. Use pointers to pass data from a read function, pass both values and pointers to a compute function, and finally pass the values to a print function.

Programs on strings:

4) Write a C program that will print out all the rotations of a string typed into it. For example the rotations of the word "space" are: space paces acesp cespa espac.(use pointers)

5) Write a program that receives the month and year from the keyboard as integers and prints the Calendar in the following format. (use pointers)

March 1995

Mon	Tue	Wed	Thu	Fri	Sat	San
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Programs on File handling in C:

6) Write a C Program to create a sequential file with at least 5 records, each record having the structure shown below:

	EMPNO.	ENAME.	BASIC.	GROSS SALARY.
Non zero positive integer.	Positive integer.	25 characters	Positive Integer	Positive Integer

Write necessary functions

a. To display all the records in the file.
b. To search for a specific record based on the EMPNO. In case the record is not found, suitable message should be displayed. Both the options in this case must be demonstrated.
c. To update basics of a employee with a given EMPNO for a particular record
d. To delete a particular record by its EMPNO.

7) Assume that master file contains three fields, Account number (numeric field), Account name (Alphabets), Amount (numeric with two decimal points for display). A transaction file contains four fields Account number (numeric field), Amount of transaction (numeric with two decimal points for display), Type of transaction (char field, W-withdrawal, D- deposits), Date of transaction (dd/mm/yy)

Write a C program to create both the files. Read records sequentially from the transaction file and update the amount in master file according to transaction type of the transaction file.

Programs on advanced C:

8) Write a C program to read a 8 bit value and check for the 5^{th} bit. If it is 1, set the 1^{st} and 2^{nd} bits; otherwise reset 1^{st} and 2^{nd} bits.

Programs on stack:

9) Write a C Program to simulate the working of a stack of integers and to perform the following operations on it:
   a. Push
   b. Pop
   c. Display

   The program should print appropriate messages for *stack overflow*, *stack underflow*, and *stack empty*.

10) Write a C Program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

Programs on recursion:

11) Write recursive C Programs for
   a. Searching an element on a given list of integers using the Binary Search method.
   b. Solving the Towers of Hanoi problem.
   c. to compute GCD of two integer numbers

Programs on queues:

12) Write a C Program to simulate the working of a queue of characters using an array. Provide the following operations:
   a. Insert
   b. Delete
   c. Display

13) Write a C Program to simulate the working of a Circular queue of integers using an array. Provide the following operations:
   a. Insert
   b. Delete
   c. Display

Programs on linked lists:

14) Write a C Program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operations to be supported are:

    a. The insertion operation
        i. At the front of a list
        ii. At the back of the list
        iii. At any position in the list

    b. Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.
    c. Searching a node based on student id and update the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.
    d. Displaying all the nodes in the list.

15) Write a C Program using dynamic variables and pointers to construct a stack of strings using singly linked list and to perform the following operations:
    a. Push
    b. Pop
    c. Display

    The program should print appropriate messages for stack overflow and stack empty.

16) Write a C program using dynamic variables and pointers to construct a queue of integers using singly linked list and to perform the following operations:
    a. Insert
    b. Delete
    c. Display

    The program should print appropriate messages for queue full and queue empty.

17) Write a C program to construct 2 ordered singly linked lists, & perform following operation.
    a. Combining two ordered list.
    b. Form a list containing the intersection of elements of two lists.

18) Write a C Program to support the following operations on a doubly linked list where each node consists of integers:
    a. Create a doubly linked list by adding each node at the front.
    b. Insert a new node to the left of the node whose key value is read as an input
    c. Delete the node of a given data, if it is found, otherwise display appropriate message.
    d. Display the contents of the list.

Programs on trees:

19) Write a C Program to construct a Binary tree & implement tree traversal methods.

Programs on sorting:

20) Write a C program to sort a given integer elements by using Selection sort.

Group Tasks:

1. Mini search Engine
2. Word Transformation
3. Expression Evaluation
4. Car garage system simulation
5. Dictionary
6. simulating calculator

General remarks:

1) Lab schedule: 3hrs/week for each student (1 hr tutorial, 2 hrs- program execution).
2) Student should complete all the lab assignments.
3) One mini project adequate per student batch (students/batch).
4) Each student should demonstrate his/her group work individually.
5) Marks Evaluation:

   Lab assignments : 40%

   Group work : 10%

Note: Develop above mentioned group work using any of the data structures.

IV SEMESTER

ADVANCED MATHEMATICS-II
(Common to all branches)

Sub Code: DMA401M **IA Marks:**
Hrs/Week: 04 **Exam Hours: 03**
Total Hrs.: 40 **Exam Marks:**

1) **Solid Geometry:** Distance formula (without proof), Division formula, Direction cosines and Direction ratios, planes and straight lines, angle between the planes

12 Hours

2) **Vectors:** Vector Algebra: Vector addition, multiplication (Dot and Cross products), Triple products Vector differentiation: Velocity , Acceleration of a vector point function, gradient, curl and divergence, solenoid and irrotational fields, simple and direct problems

12 Hours

3) **Laplace transforms :** Definitions, Laplace transforms of elementary functions, derivatives and integrals. Inverse transforms, Applications . Laplace transforms to differential equations.

16 Hours

Text Books:

1) Higher Engineering Mathematics – B S Grewal
2) Higher Engineering Mathematics – H K Dass

IV SEMESTER

ENGINEERING MATHMATICS - IV

Sub Code: UMA401C **IA Marks:**
Hrs/Week: 04 **Exam Hours: 03**
Total Hrs.: 52 **Exam Marks:**

Unit-I

COMPLEX VARIABLES:
Functions of complex variables, Limit, continuity and differentiability (definitions only), Analytic functions, Cauchy-Reimann equations in Cartesian and Polar forms - consequences, construction of analytic function (Cartesian and polar forms), Definition of Conformal transformations: Z^2, e^z and $z + a^2/z$ *(z#0)* Bilinear transformations.

Complex Integrations**:**
Line integral, Cauchy's theorem - corollaries, Cauchy's integral formula. Taylor and Laurent's series (statements only), Singularities, Poles, Calculation of Residues, Residue theorem (Without proof) - problems.

14 Hours

Unit-II

SPECIAL FUNCTIONS:
Series solution of Bessel's differential equation, recurrence formulae, enerating function, orthogonal property, Bessel's integral formula. Series solution of Legendre's differential equation, Recurrence formulae, Generating function, orthogonal property, Rodrigue's formula.

14 Hours

Unit-III

STATISTICS AND PROBABILITY:

Curve fitting by the method of least squares: $y = a + bx$, $y = ab^x$, $y = a + bx + c X^2$. Correlation and Regression. Probability - addition rule, conditional probability, multiplication rule, Bayes' rule.
Discrete and continuous random variables-PDF-CDF, Binomial, Poisson, Exponential and Normal distributions

12 Hours

Unit-IV

SAMPLING DISTRIBUTION:

Sampling, Sampling distribution, Standard error, Null and alternate hypotheses, Type I and Type II errors, Testing of hypothesis for Means, Level of Significance for Means, Confidence limits for Means, large and small samples, Student's *t*-distribution.

JOINT PROBABILITY DISTRIBUTION AND MARKOV CHAINS:

Concept of joint probability, Joint distributions - discrete random variables, Independent random variables, Problems on expectation and variance. Markov chains - Introduction, Probability vectors, Stochastic Matrices, Fixed Points and Regular stochastic Matrices, Markov chains, higher transition probabilities, stationary distribution of regular Markov chains and absorbing states.

12 Hours

TEXT BOOKS:
Higher Engineering Mathematics (36th edition-20OZ) by Dr. B S Grewal, Khanna Publishers, New Delhi.

Unit-I: Complex Variables:

Chapter 20: 20.1 to 20.6, 20.8(4), 20.9(1), 20.10(1 to 3)
20.12 to 20.14, 20.16(2,3) 20.17 *to* 20.19

Unit-II: Special Functions:

Chapter 16: 16.6 to 16.9, 16.11, 16.13 to 16.17

Unit-III: Statistics and Probability:

Chapter 1: 1.13,1.14
Chapter 23: 23.9, 23.10, 23.14, 23.16 to 23.21,
23.26, 23.27(5), 23.28

Unit-IV: Samplin2 Distributions:

Chapter 23: 23.31 to 23.33, 23.34 to 23.36
Joint Probability distributions and Markov Chains:

TEXT BOOKS:
Theory and Problems of Probability by Seymour Lipschutz (Schaum' Series) - Relevant articles of Chapter 5 and chapter 7.

REFERENCE BOOKS:
Advanced Engineering Mathematics by E Kreyszig (John Wiley & Sons)

Unit-I	Functions of Complex Variable – Bilinear transformation Complex integration	One Question
Unit-II	Bessel's Differential	One Question
	Equation Legendre's Differential Equation	One Question
Unit-III	Curve fitting - Bayes' Rule Random Variables - End	One Question
Unit-IV	Sampling Distributions Joint Probability Distributions- End	One Question

MICROPROCESSORS

Sem	**: IV**	**Credits**	**: 4**
Sub code	**: UIS402C**	**Hrs/week**	**: 4**
		Total hrs	**: 52**

UNIT I

An Introduction to Computers, Microcomputers, and Microprocessor
Types of computers; Overview of microcomputer structure and operation; Microprocessor evolution and types; The 8086 family overview-8086 Internal architecture; Introduction to programming 8086

Introduction to 8086 Family Assembly Language Programming
Program development steps; Writing programs for use with an assembler; Assembly language program development tools; Constructing machine codes for 8086 instruction; MOV instruction coding examples

Implementing Standard Program Structure In 8086 Assembly Language
Simple sequence programs; Jumps, Flags, and Conditional Jumps; If-Then, If-Then-Else, and Multiple If-Then-Else programs; While-Do programs; Repeat-Until Programs; Instruction timing and delay loops

13 Hours

UNIT II

8086 Instruction Description and Assembler Directives
Instruction descriptions; Assembler directives

Strings, Procedures, and Macro
The 8086 string instructions; Writing and using procedures; Writing and using assembler macros.

13 Hours

UNIT III

Using Assembly Language with C/C++
Using assembly language with C/C++ for 16-bit applications.

8086 Interrupts and Interrupt Applications
8086 Interrupts and interrupt responses; Hardware Interrupt applications; 8254 software-programmable timer/counter; 8259A priority interrupt controller; software interrupt applications.

8086 System Connections and Timing
A basic 8086 microcomputer system; 8086 minimum and maximum modes; Addressing memory and ports in microcomputer systems; 8086 addressing and address decoding; How the 8086 microprocessor accesses memory and ports; 8086 timings.

13 Hours

UNIT IV

Digital Interfacing and Analog Interfacing

Programmable parallel ports and handshake input/output; Interfacing microprocessor to keyboards; Interfacing to alphanumeric displays; Review of operational amplifier characteristics and circuits; Sensors and Transducers; D/A operation, interfacing; A/D specifications, interfacing

Direct Memory Access, Coprocessors

Basic DMA operation; The 8237 DMA controller; A Coprocessor- The 8087 math coprocessor.

The Pentium II, Pentium III, and Pentium IV Microprocessors

Introduction to Pentium II microprocessor; Pentium II software changes; The Pentium III; The Pentium IV.

13 Hours

Note: Every student has to do one assignment comparing atleast 4 processors belonging to different families.

Text Books:

1. **Microprocessor and Interfacing,** Douglas V. Hall, Revised 2nd Edition, TMH, 2006.(Chapters 2, 3,4,5,6,7,8,9,10,11)

2. **The Intel Microprocessors,** 6th Edition, Barry B. Brey, Pearson/PHI 2006.(Chapters 8,13, 19)

Reference Books:

1. **Advanced Microprocessors and IBM-PC assembly Language programming,** K. Udaya Kumar and B.S. Umashankar, TMH 2003.
2. **The Intel Microprocessor Family: Hardware and Software Principles and Applications,** James L. Antonakos, Thomson, 2007.

ANALYSIS AND DESIGN OF ALGORITHMS

Sem	**: IV**	**Credits**	**: 4**
Sub code	**: UIS403C**	**Hrs/week**	**: 4**
		Total hrs	**: 52**

UNIT I

Introduction: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures **6 Hours**

Fundamentals of the Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Nonrecursive and Recursive Algorithms, Example –Fibonacci Numbers **7 Hours**

UNIT II

Brute Force: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search **3 Hours**

Divide and Conquer: Mergesort, Quicksort, Binary Search , Binary tree traversals and related properties, Multiplication of large integers and Stressen's Matrix Multiplication. **5 Hours**

Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Decrease by constant factor Algorithms,Variable –size-decrease algoruthms **5 Hours**

UNIT III

Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heapsort, Problem Reduction **5 Hours**

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching, Hashing, B-trees. **4 Hours**

Dynamic Programming: Computing a Binomial Coefficient, Warshall's Algorithm, : Floyd's Algorithms, The Knapsack Problem and Memory Functions **4 Hours**

UNIT IV

Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees, Knapsack Problem. **4 Hours**

Limitation of Algorithm Power: Lower-Bound arguments, Decision Tree,P,NP,NP-Complete Problem. **4 Hours**

Coping with the Limitations of algorithm power : Backtracking And Branch & Bound techniques Approximation Algorithm for NP-Hard problems. **5 Hours**

Text Book

1. **Introduction to The Design & Analysis of Algorithms** , Anany Levitin, 2nd Edition, Pearson ducation, 2007. (Chapter 1, 2.1 to 2.5, 3.1, 3.2, 3.4, 4.1 to 4.5, 5.1 to 5.4, 6.1, 6.3, 6.4, 6.6, 7.1 to 7.3, 8.1, 8.2, 8.4, 9, 11.1, 11.2, 11.3, 12.1, 12.2, 12.3).

Reference Books

1. **Introduction to Algorithms** , Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 2nd Edition, PHI, 2006.
2. **Computer Algorithms** by Horowitz E., Sahni S., Rajasekaran S., Galgotia Publications, 2001.

Computer Organization

Sem	**: IV**	**Credits**	**: 4**
Sub code	**: UIS404C**	**Hrs/week**	**: 4**
		Total hrs	**: 52**

Unit I

.Basic Structure of Computer**:** Computer Types. Functional Units, Basic Operational Concepts, Bus Structures, Performance – processor clock, Basic Performance Equation, Clock rate, Performance Measurement, Machine Instructions and Programs**:** Numbers, Arithmetic Operations and Characteristics, Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing. Addressing Modes, Assembly language, Basic Input and Output operations, Stacks and Queues, Subroutines.

13 HRS

Unit II

Additional Instructions, Encoding of machine instructions. Input/Output organization**:** Accessing I/O Devices, Interrupts-interrupt hardware, Enabling and disabling interrupts, Handling multiple devices, Controlling device requests, Exceptions, Direct memory access, Buses, Interface circuits, Standard I/O interfaces-PCI Bus, SCSI Bus, USB.

13 HRS

Unit III

Memory system**:** Basic Concepts, Semiconductor RAM Memories, Read Only Memories, Speed, Size and cost, Cache Memories, Mapping Functions, Virtual memories, Basic Arithmetic concepts for ALU**:** Addition and subtraction of signed numbers, Design of fast adders, Multiplication of positive numbers, Signed operand multiplication, Fast multiplication, Integer division.

13 HRS

Unit IV

Floating point numbers and operations, Basic processing unit: Fundamental concepts, Execution of a complete instruction, Multiple bus organization, Hard-wired control, Micro programmed control, Advanced concepts: Pipelining, role of cache memories, pipeline performance.

13 HRS

Text Book:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 5th edition, TMH, 2002. (Chapter 1.1 to 1.4, 1.6.1, 1.6.2, 1.6.4, 1.6.7, 1.8, Chapter 2.1 to 2.10, 2.12, Chapter 4.1, 4.2.1 to 4.2.5, 4.4 to 4.7, Chapter 5.1 to 5.4, 5.5, 5.5.1, 5.6, 5.7, 5.9, Chapter 6, 7, Chapter 8.1)

Reference Books:

1. Computer Organization and Architecture, William Stallings,7th edition,PHI,2006
2. Computer Systems Design and Architecture, Vincent P.Heuring & Harry F.Jordan, 2nd edition, Pearson Education, 2004.

FILE AND INFORMATION STRUCTURES

Sem	**: IV**	**Hrs/Week**	**: 4Hrs**
Sub Code	**: UIS405C**	**Credits**	**: 4**
		Total Teaching Hours	**: 52**

UNIT I

INTRODUCTION TO THE DESIGN OF FILE STRUCTURE: The Heart of the file structures Design, A Short History of File Structures Design; Fundamental File Processing Operations*:* Physical Files and Logical Files, Opening Files, Closing Files, Reading and Writing, Seeking; Secondary Storage: Disks, the organization of disks, estimating capacities space needs, organizing tracks by sector, organizing tracks by blocks, non-data overhead, the cost of a disk access; *CD-ROM:* Physical Organization of CD-ROM, CD-ROM Strengths and Weaknesses. FUNDAMENTAL FILE STRUCTURE CONCEPTS: Field and Record Organization, Buffer Management. Using Classes to Manipulate Buffers, Using Inheritance for Record Buffer Classes, Record access, Header Records. **13 hours**

UNIT II

ORGANIZATION OF FILES FOR PERFORMANCE: Reclaiming Space in files, Internal Sorting and Binary Searching, Key sorting . INDEXING: Introduction, A Simple Index for Entry - Sequenced File, Object-Oriented support for Indexed, Entry-Sequenced Files of Data Objects, operation required to maintain an indexed file, Class text index file, Indexes that are too large to hold in Memory, Indexing to provide access by Multiple keys, Retrieval Using Combinations of Secondary Keys, Improving the Secondary Index structure, Inverted List.

13 hours

UNIT III

COSEQUENTIAL PROCESSING & THE SORTING OF LARGE FILES: An object oriented model for implementing consequential processes: Matching Names in two lists, Merging two lists, Summary of the consequential processing model, Extension of the model to include multiway merging : A K-way merge algorithm, a selective tree for merging large numbers of lists, A second look at sorting in memory: overlapping processing and I/O: heapsort, Building the heap while reading the file, sorting while writing to the file, Merging as a way of sorting large files on disk: Time for merge sort , Sorting a file that is ten times larger , the cost of in creasing the file size, Hardware based improvements, Decreasing the number of seeks using multiple step merges, Increasing run length s using replacement selection, replacement selection plus multistep merging, Using two disk drives with replacement selection , More drives, More processors, Effects of multiprogramming, A conceptual toolkit for external sorting.

MULTI-LEVEL INDEXING AND B-TREES: Statement of the problem, Indexing with BST, AVL tree, paged binary tree, problem with paged binary trees. Multilevel indexing: A better approach to tree indexes, Working up from the bottom, Example of Creating a B-Tree, An Object-Oriented Representation of B-Trees, B-Tree Methods search, insert & others, Formal Definition of B-Tree Properties, Worst-case Search Depth, Deletion, Merging and Redistribution during insertion.

13 hours

UNIT IV

INDEXED SEQUENTIAL FILE ACCESS AND PREFIX B +TREES :Indexed Sequential Access, Maintaining a Sequence Set, Adding a Simple Index to the Sequence Set, The Content of the Index*:* Separators Instead of Keys, The Simple Prefix B+ Tree and its maintenance, Index Set Block Size, Internal Structure of Index Set Blocks*:* A Variable-order B- Tree, Loading a Simple Prefix B+ Trees. HASHING: Introduction, A Simple Hashing Algorithm, Hashing Functions and Record Distribution, Extra Memory used, Collision resolution by progressive overflow, Sorting more than one record per address Buckets, Making deletions, Other collision resolution techniques. **13 hours**

TEXT BOOKS:

1. Michael J. Folk, et al: File Structures-An Object Oriented Approach with C++ (Third Edition) Pearson education 2004 (Chapters 1.1, 1.2, Chapter2.1to2.5, Chapter 3.1, 3.4, 3.5, 3.6, Chapter 4.1, 3.9, 4.2, 4.3, 5.1, 5.2.2, Chapter 6.2 to 6.4, Chapter 7.1 to 7.8, Chapter 8.1,8.3 to 8.5,Chapter 9.1 to 9.8, 9.10 to 9.13, Chapter 10.1 to 10.9, Chapter 11.1 to 11.8)

REFERENCE BOOKS:

1. Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.
2. File Structures: Conceptual Toolkit with C++, Venkatesh N.Baitipuli, 1st Edition, 2004
3. Introduction Algorithms: Thomas H. Cormen, et.al, 2nd edition, PHI

Theoretical Foundations of Computer Science

Sem	**: IV**	**Credits**	**: 3**
Sub code	**: UIS406C**	**Hrs/week**	**: 3**
		Total hrs	**: 42**

Unit-I

Introduction to Finite Automata: Introduction to Finite Automata; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata.

Finite Automata, Regular Expressions: An application of finite automata; Finite automata with Epsilon-transitions; Regular expressions; Finite Automata and Regular Expressions; Applications of Regular Expressions. **11Hours**

Unit-II

Regular Languages, Properties of Regular Languages: Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata.

Context-Free Grammars And Languages: Context–free grammars; Parse trees; Applications; Ambiguity in grammars and Languages. **11Hours**

Unit-III

Pushdown Automata: Definition of the Pushdown automata; The languages of a PDA; Deterministic Pushdown Automata.

Properties of Context-Free Languages: Normal forms for CFGs; The pumping lemma for CFGs; **10Hours**

Unit-IV

Introduction To Turing Machine: Problems that Computers cannot solve; The turning machine; Programming techniques for Turning Machines; Extensions to the basic Turning Machines; Turing Machine and Computers. **10Hours**

Text Books:

1. John E.. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson education, 2007. (Chapters: 1.1, 1.5, 2.2 to 2.5, 3.1 to 3.3, 4, 5, 6.1, 6.2, 6.4, 7, 8.1 to 8.4, 8.6)

Reference Books:

1. Peter. Linz: An Introduction to Formal Languages and Automata", Third Edition, Fifth printing
2. Raymond Greenlaw, H.James Hoover: Fundamentals of the Theory of Computation,
1. Principles and Practice, Morgan Kaufmann, 1998.
2. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.
3. Daniel I.A. Cohen: Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, 2004.
4. Thomas, A., Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition, Pearson Education, 2006
5. John, E., Hopcroft, Jeffrey D.Ullman: “Introduction to Automata Theory, Languages and Computation”, Narosa.

FILE AND INFORMATION STRUCTURES LAB

Sem	**: IV**	**Hrs/Week**	**: 3Hrs**
Sub Code	**: UIS407L**	**Credits**	**: 1.5**

1. **Program on Text File handling.**
   Write a C++ program to create text file with each record having the structure shown below.
   Student USN: 10 characters.
   Student name: 15 characters.
   Student Address: 10 characters.
   Student Course: 05 characters.
   Student Branch: 05 characters.
   Write necessary methods to Read, Write & Display the records.

2. **Program on Binary File handling.**
   Write a C++ program to create Binary file with each record having the structure shown below.
   Employee ID: Alphanumeric
   Employee Phone Number: Positive integer.
   Employee Basic Salary: Positive integer.
   Write necessary methods to Read, Write & Display the records.

3. **Program on file handling using delimited field, variable length (using length indicator) record structure.**
   Write a C++ program to create file with delimited fields, variable length record having the fields shown below.
   Customer ID: 10 Characters
   Customer Name: 10 characters.
   Customer Address: 25charactres.
   Write necessary methods to Read, Write & Display the records.

4. **Program on file handling using delimited record structure with each field begins with length indicator**.
   Write a C++ program to create file with delimited record structure where each field shown below begins with length indicator,
   Teacher ID: Non-zero integer.
   First Name: 10 Characters.
   Last Name: 10 Characters.
   Details: 15 Characters.
   Write necessary methods to Read, Write & Display the records.

5. **Program on file handling using fixed field, fixed length record structure.**
   Write a C++ program to create file with fixed field, fixed length record structure, having the fields shown below
   Subject Code: 10 characters.
   Subject name: 15 characters.
   Faculty: 10 characters.
   Write necessary methods to Read, Write & Display the records.

6. **Program on Key sorting technique.**
   Write a C++ program to create file, with each record having the structure shown below.
   Item Number: +ve integer
   Item Name: 10 characters
   Item type: 10 characters.
   Item price: float
   Quantity: integer
   Write necessary methods to Read , Write & Display the records & method to sort the records with Item Number as the key using key sorting technique.

7. **Program to search a record using an Index.**
   Write a C++ program to create file with each record having the structure shown below.
   Department Name: 15 characters.
   No of students: integer.
   No of teaching staff: integer.
   No of Non- teaching staff: integer.

   Develop an index for the file with the department Name as the primary key.
   Write necessary methods to Read, Write the records in the file & to display the record by searching an index.

8. **Program to search a record using Secondary Index.**
   Write a C++ program to create file with each record having the structure shown below.
   Patient Number: 10 characters.
   Patient Name: 10 Characters.
   Disease: 15 characters.

   Develop a Secondary index for the above file with the patient name as the Secondary key.
   Write necessary methods to Read, Write the records in the file & to display the record by searching secondary index.

9. **Program to search a record using inverted list.**
   Write a C++ program to create file with each record having the structure shown below.
   Book number: integer.
   Title of the book: 10 characters.
   Author:10 characters
   Publication: 10 characters.

Develop an Secondary index for the above file with the author as the Secondary key.
Develop an inverted list to rearrange secondary key index when insertion of new record is to be done. Write necessary methods to Read, Write the records in the file & to display the record using inverted list.

10. Program to search a record using combination of secondary keys.

Write a C++ program to create file with each record having the structure shown below.

Record label: 5 characters.
Record ID: integer.
Title: 10 characters.
Composer: 10 characters

Develop a composer & title index for the above file.

Write necessary methods to Read, Write the records in the file & to search & display the record using combination of secondary keys(composer & title).

11. Program on co-sequential processing (matching)

Write a C++ program to create file with each record having the structure shown below.

Student USN: 10 characters.
Student name: 15 characters.
Student Address: 10 characters.
Student Course: 05 characters.
Student Branch: 05 characters.

Develop master transaction process for each student record (master) to print the student Information & list of all credits (transaction) taken by the student using matching operation.
Write necessary methods to Read , Write the records in the master & transaction file & display the records.

12. Program on consequential processing (merging)

Write a C++ program to create file with each record having the structure shown below.

Account Number: 10 characters.
Account holder name: 10 characters
Account type: 05 Characters.
Balance: float.

Sort the file using 4-way merge and demonstrate the phases of merging using heapsort.

13. Program to search a record using B tree.

Write a C++ program to create file with each record having the structure shown below.

Game Name: 10 characters.
Game type: 10 characters.
No of players: integer.

Create a B Tree for the above file with the game name as key.
Write the methods to read & write the records in the file & to search record using B Tree.

14. Program to search a record using B+ tree.

Write a C++ program to create file with each record having the structure shown below.

Car model Number: 10 characters.
Car price: float.
Vendors: 10 characters.

Create a B+ Tree for the above file with the car model Number as key.
Write the methods to read & write the records in the file & to search record using B+ Tree.

15. Program on Hashing.

Write a C++ program to create file with each record having the structure shown below.

Mobile model Number: 10 characters.
Mobile phone Number: integer.
Mobile holder name: 10 characters

Develop a hashed index of the above file with Mobile model Number as a key.
Write the methods to read & write the records in the file & to search record using hashed index.

ANALYSIS OF ALGORITHMS LABORATORY

Semester	**: IV**	**Credits**	**: 1.5**
Code	**: UIS408L**	**Teaching hours/week**	**: 3**

1. Consider the following standard time complexity functions:
   F(n)=n, log n , n2, n3, 2n , n!
   Write a program to that takes input value 'n' and find the value of F(n) by varying I from 1 to n
   Note : Use switch statement to consider different functions.
   Draw the graph of **Time versus 'Input n'** to display these results, and observe the effect of increasing the n value for each function.

2. Write a program to generate 'n' random numbers and sort them using bubble sort. Perform apriority and posteriori analysis (time complexity) in following cases.
   a. n= 5, and the value of each number is lesser than 100
   b. n=10000, and the value of each number is lesser than 10000
   Analyze the memory requirements in each case.

3. Write a program for recursive binary search and Iterative Binary search in an a array of N elements. Perform apriority and a posteriori analysis of the time required to search an element in following cases.
   a. N=10 (Manually Entering inputs).
   b. N=5000.
   c. N= 50000.

4. How many comparisons (both successful and unsuccessful) will be made by the brute-force string-matching algorithm in searching for each of the following pattern in the binary text of 1000 zeros and/or ones ?
   a. 00001 b. 10000 c. 01010

5. Apply Quick sort method to sort elements of array in following cases. Observe the best case and worst case time complexity.
   An array with 1000 elements
   An ordered array (increased or decreased) with 1000 elements

6. Sort the contents of a text file using insertion sort & hence find the time required to sort the contents.

7. Obtain the topological ordering of a vertices in a given digraph using DFS.

8. a. Implement 0/1 Knapsack problem using Dynamic Programming.
   b. Find the Binomial Co-efficient using Dynamic Programming.

9. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkistra`s algorithm.

10. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal`s Algorithm.

11. Print all the nodes reachable from a given starting node in a digraph using Breadth First Search method.

12. Sort a given set of integer elements using Heap Sort and hence find the time complexity.

13. Implement Horspool Algorithm for string matching.

14. Implements All pair shortest path problem using Floyd`s algorithm.

15. Implements N Queen`s problem using Backtacking.